Getting started with the BLUEMICROSYSTEM1 Bluetooth low energy and sensor software expansion for STM32Cube

Introduction

This user manual describes how to get started with the BLUEMICROSYSTEM1 software.

BLUEMICROSYSTEM1 provides a complete middleware for STM32 to build applications using inertial (e.g. motion MEMS) and environmental (e.g. humidity, pressure, temperature) sensors, leveraging Bluetooth Low Energy for sensor data communication. The package includes:

- the osxMotionFX (INEMOEngine PRO under OPEN.MEMS license), which uses advanced algorithms to integrate outputs from multiple MEMS sensors in a "smart" way, independent of environmental conditions, to reach optimal performance.
- the osxMotionAR (iNEMOEngine PRO under OPEN.MEMS license) real-time activity-recognition algorithm
- osxMotionCP (iNEMOEngine PRO under OPEN.MEMS license) carry-position recognition algorithm

All of which are easily portable across different MCU families thanks to STM32Cube.

The package is compatible with the BlueMS Android®/iOS® application (Version > 2.0.0) available on respective play store/iTunes market. That application can be used for displaying information sent by the Bluetooth Low Energy protocol.

The software provides implementation examples for STM32 Nucleo platforms (NUCLEO-F401RE and NUCLEO-L476RG) equipped with the X-NUCLEO-IKS01A1 expansion board featuring temperature, humidity, pressure and motion MEMS sensors, further mounted with the X-NUCLEO-IDB04A1or X-NUCLEO-IDB05A1 expansion boards, which enable the Bluetooth protocol.

The software is based on STM32Cube technology. Information about STM32Cube is available on www.st.com at http://www.st.com/stm32cube.

Contents

List of figures

1 BLUEMICROSYSTEM1 software description

1.1 Overview

The key features of the BLUEMICROSYSTEM1 package are:

- Complete middleware to build applications using temperature and humidity sensors (HTS221), pressure sensor (LPS25HB) and motion sensors (LIS3MDL and LSM6DS0)
- Compatible with the motion sensor LSM6DS3 DIL24 expansion component
- Very low power Bluetooth Low Energy (BlueNRG) single-mode network processor, compliant with Bluetooth specification core 4.0 for transmitting information to one client
- osxMotionFX (iNEMOEngine PRO) real-time motion-sensor data fusion (under OPEN.MEMS license) that combines the output from multiple MEMS sensors
- osxMotionAR activity recognition middleware (iNEMOEngine PRO) under open.MEMS license)
- osxMotionCP carry position recognition middleware (iNEMOEngine PRO) under open.MEMS license)
- Easy portability across different MCU families, thanks to STM32Cube
- Sample implementation available on board X-NUCLEO-IKS01A1 and X-NUCLEO-IDB04A1/X-NUCLEO-IDB05A1, when both connected to one NUCLEO-F401RE

The software runs on the STM32 and includes drivers that recognize the Bluetooth Low Energy (BlueNRG) and four sensor devices (HTS221, LPS25HB, LSM6DS0, LIS3MDL).

This package is compatible with the BlueMS Android®/iOS® application (Version > 2.0.0) available on respective play store/iTunes market. That application can be used for displaying information sent by the Bluetooth Low Energy protocol.

1.2 Architecture

The software is based on the STM32CubeHAL, the hardware abstraction layer for the STM32 microcontroller.

The package provides a board support package (BSP) for the sensor expansion board and some middleware components for serial communication with a PC.

The software layers used by the application software to access and use the sensor expansion board are:

- STM32Cube HAL layer: The HAL driver layer provides a generic multi-instance simple set of APIs (application programming interfaces) to interact with the upper layers (application, libraries and stacks). It is composed of generic and extension APIs. It is directly built around a generic architecture and allows the layers that are built upon, such as the middleware layer, to implement their functionalities without dependency on the specific hardware configuration for a given microcontroller unit (MCU). This structure improves library code reusability and guarantees easy portability to other devices.
- Board support package (BSP) layer: The software package must support the peripherals on the STM32 Nucleo board independent of the MCU. This software is included in the board support package (BSP). This is a limited set of APIs which provides a programming interface for certain board specific peripherals, e.g. the LED, the user button, etc. This interface also helps in identifying the specific board version. For the sensor expansion board, it provides the programming interface for various

inertial and environmental sensors. It also provides support for initializing and reading sensor data.

The diagram below illustrates the software architecture of the package:
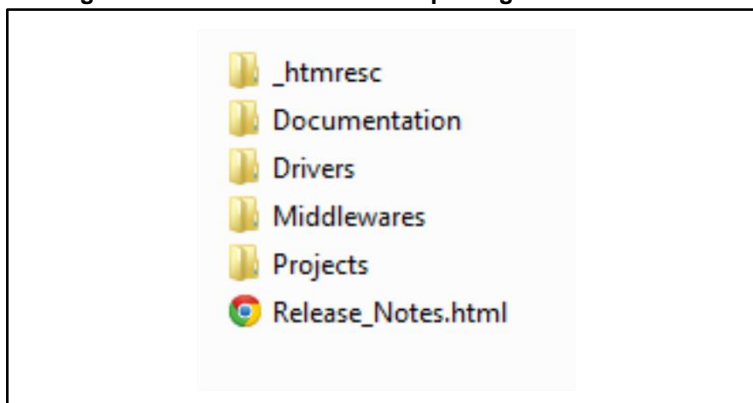
Figure 1: BLUEMICROSYSTEM1 overall system architecture
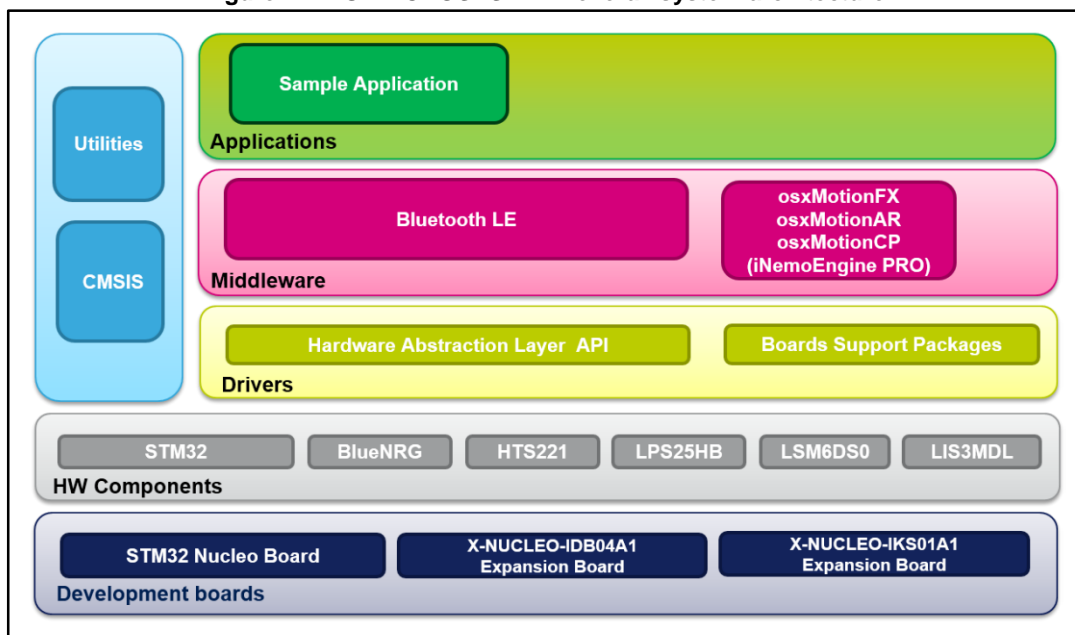
1.3 Folders structure

The folder structure of the package in shown below.

Figure 2: BLUEMICROSYSTEM1 package folder structure

The following folders are included in the software package:

- **Documentation**: contains a compiled HTML file generated from the source code and documentation that details the software components and APIs.
- **Drivers**: contains the HAL drivers, the board-specific drivers for each supported board or hardware platform, including the on-board components and the CMSIS vendor-independent hardware abstraction layer for the Cortex-M processor series.
- **Middleware**: contains libraries and protocols for BlueNRG Bluetooth Low Energy, for osxMotionFX sensor fusion library, for osxMotionAR real time activity recognition library and for osxMotionCP carry position recognition library(iNEMOEngine PRO).

- **Projects**: contains a sample application used for transmitting the output of the sensor data and the output of the osxMotionFX, osxMotionAR, osxMotionCP sensor fusion libraries by using the Bluetooth low energy protocol, provided for the NUCLEO-F401RE and NUCLEO-L476RG platforms with three development environments (IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM), System Workbench for STM32).

1.4 APIs

Detailed technical information about the APIs available to the user can be found in a compiled HTML file located inside the "Documentation" folder of the software package, where all the functions and parameters are described.

1.5 Sample application description

An example application using the X-NUCLEO-IKS01A1 and X-NUCLEO-IDB04A1 expansion boards with the NUCLEO-F401RE board is provided in the "Projects" directory. Ready-to-build projects are available for multiple IDEs.

This application reads the sensor data values from the accelerometer, magnetometer and gyroscope at a 100 samples/second frequency. The osxMotionFX (iNEMOEngine PRO) library combines these data sensor values to produce quaternions at the same rate of 100 quaternions/second, which are transmitted to the client connected through the Bluetooth Low Energy protocol to display real movement using a vendor-specific BLE service. There are two defines in the `osx_bms_config.h` file that control how many quaternions the application sends to the Bluetooth client:

- `QUAT_UPDATE_MUL_10MS`: defines the transmission rate for each set of quaternions by multiples of 10 ms.
- `SEND_N_QUATERNIONS`: defines how many quaternions are sent on each Bluetooth package.

By default, the application sends three quaternions every 30 ms.

In the same `osx_bms_config.h` file, there are a further 7 defines:

- `ENV_UPDATE_MUL_10MS` defines the transmission rate for Temperature/Pressure and Humidity
- `ACC_GYRO_MAG_UPDATE_MUL_10MS` defines the transmission rate for Accelerometer/Gyroscope and Magnetometer values
- `QUAT_FLOAT_UPDATE_MUL_10MS` defines the transmission rate for the float precision quaternions
- `OSX_BMS_DEBUG_CONNECTION` and `OSX_BMS_DEBUG_NOTIFY_TRAMISSION` for enabling certain debugging information for BLE communication
- `OSX_BMS_MOTIONAR` for enabling the osxMotionAR activity recognition algorithm
- `OSX_BMS_MOTIONCP` for enabling the osxMotionCP carry position recognition algorithm

The osxMotionFX (iNEMOEngine PRO) library has an autocalibrating procedure and the calibration status is transmitted by BLE to the client. By pressing the user button on the NUCLEO-F401RE board, it is possible to reset the library calibration status, thereby forcing a new autocalibration procedure.

The osxMotionAR (iNEMOEngine PRO) library is able to recognize the following activities: stationary, walking, fast walking, jogging, biking and driving. The osxMotionAR can be enabled by defining the `OSX_BMS_MOTIONAR` definition in the `osx_bms_config.h` file.

The osxMotionCP (iNEMOEngine PRO) library recognizes and provides real-time information about how the user is carrying the board; i.e., the phone carrying position: on desk, in hand, near head, shirt pocket, trouser pocket and arm swing. The osxMotionCP can be enabled by defining the `OSX_BMS_MOTIONCP` define in the `osx_bms_config.h` file.

Using a terminal, the user can control the initialization phase of the System, the osxMotionFX/osxMotionAR license authentication and calibration status, and Free Fall detection if the LSM6DS3 DIL24 expansion component is present. Similarly, the user can control all the subscriptions to the notify services and all the sensor values (temperature/Humidity/Pressure) sent via bluetooth.

1.6 Android and iOS sample client application

The BLUEMICROSYTEM1 software for STM32Cube is compatible with the BlueMS Android/iOS applications (Version > 2.0.0) available at the respective play/iOS stores.

This section describes the BlueMS application, using the Android version as the example. Before using this utility, the user must ensure that the necessary drivers, as explained in the chapter dedicated to system setup, are installed and the expansion boards along with STM32 Nucleo board is connected to the PC.

Using one terminal (setting the UART port to 460800 bps, 8 bit, No Parity,1 stop bit), it is possible to control the initialization phase.

Figure 3: Terminal output after the intialization

Following initialization, you will see:

- The Nucleo board being used:
  - STM32L476RG-Nucleo board
- When the STM32 Nucleo Code was compiled and with which IDE

- Compiled Oct 8 2015 09:04:32 (IAR)
- Quaternions short precision (and number of quaternions sent on each package) transmission frequency:
  - Send 3 quaternions every 30 ms
- Quaternions float precision transmission frequency:
  - Send float precision quaternion every 20 ms
- Temperature/Presure/Umidity transmission frequency:
  - Send Temperature/Humidity/Pressure every 500 ms
- Accelerometer/Gyroscope and Magnetometer transmission frequency:
  - Send Acc/Gyro/Magneto every 100 ms
- If we have enabled the BlueNRG debug messages:
  - Debug connection enabled debug notify transmission enabled
- If the Bluetooth server is initialized, which BlueNRG board is being used with which HW and SW version, and the with which name/MAC the board is visible:
  - SERVER: BLE Stack Initialized — Board type=IDB05A1 HWver=49, FWver=1811 - BoardName= BlueMS1, BoardMAC = aa:aa:aa:dd:ee:ff:
- If the Bluetooth service that exposes all the HW characteristics is added:
  - HW Service W2ST added successfully
- If the Bluetooth service that exposes all the SW characteristics is added:
  - SW Service W2ST added successfully
- If the Bluetooth service for enabling the console (stdout/stdin) is added:
  - Console Service W2ST added successfully
- If the Bluetooth service for sending configuration commands to Nucleo is added:
  - Config Service W2ST added successfully
- If the X-NUCLEO-IKS01A1 board sensors are initialized:
  - OK Temperature Sensor
  - OK Pressure Sensor
  - OK 6 AxesSensor
  - OK Magneto Sensor
- If LSM6DS3 DIL24 is present DS3 on board
- If the osxMotionFX license was initialized correctly:
  - Enabled ST osxMotionFX x9/x6 v1.0.7
- If the osxMotionAR license was initialized correctly:
  - Enabled ST osxMotionAR V1.0.1
- If the osxMotionCP license was initialized correctly:
  - Enabled ST osxMotionCP V1.0.0
- Whether or not the calibration is already present in RAM or FLASH memory:
  - Calibration not present in RAM

On the same terminal, during the connection phase you will see:

- The MAC address of the connected Android/iOS device (green arrow in the following figure)
- When the board receives a message for resetting the calibration or when a new Magneto calibration is achieved (blue sections in the following figure)
- When the Client wants to subscribe or unsubscribe to certain BLE characteristics
- The values of the Temperature/Humidity/Pressure sent to the client (only if their value is changed with respect the last transmitted value)
- If the LSM6DS3 is present, it will generate an interrupt if it detects a Free Fall (yellow arrow in the following figure)

Figure 4: Terminal output during a normal connection

The first time that the application starts, a popup window warns that it is only compatible only with BLUEMICROSYTEM1 version > 2.0. Pressing the download button, you may be forwarded on the correct page on www.st.com for downloading the new firmware package for NUCLEO-F401RE or NUCLEO-L476RG.

Figure 5: Initial application start
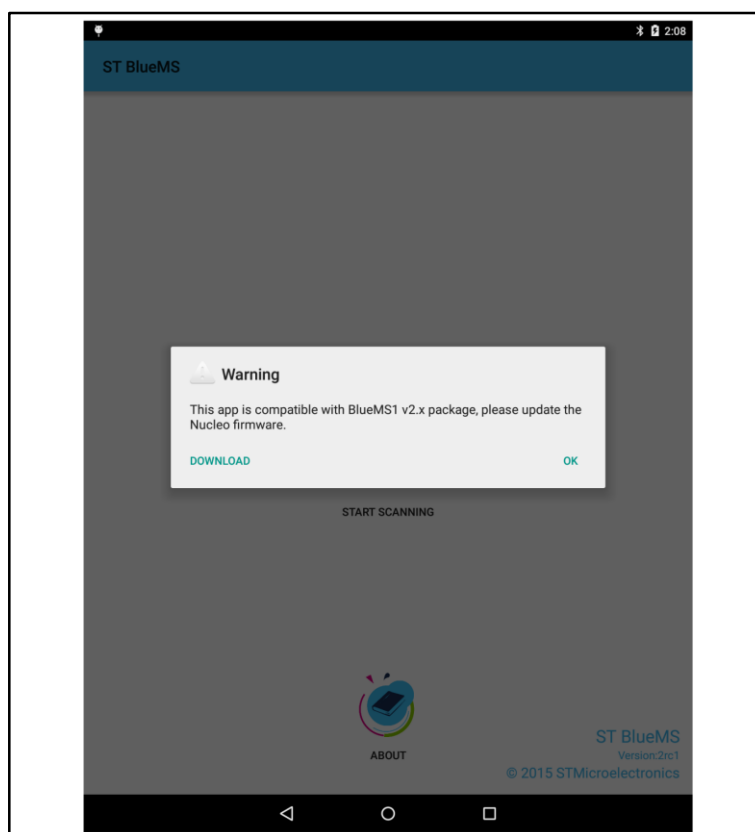
It is then possible to start the Bluetooth Low energy scan to search for compliant devices, and subsequently choose the device to connect.

**Figure 6: Android application device scanning**
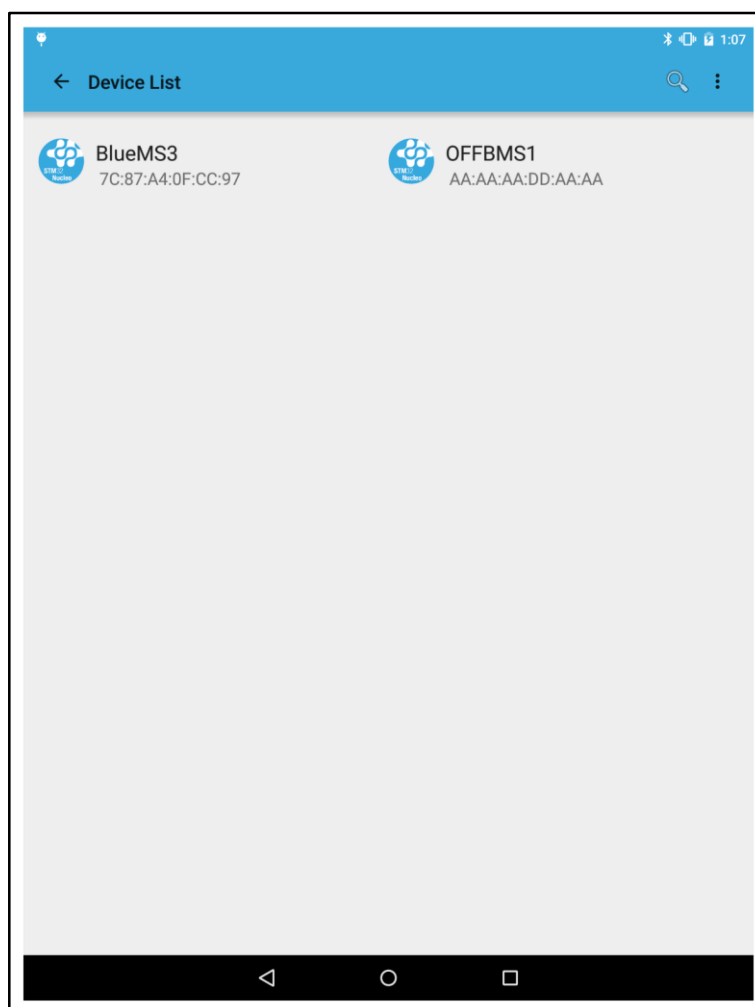
The first page shown after the connection is the environmental page with the Temperature/Pressure and Humidity information measured by the X-NUCLEO-IKS01A1 board sensors (the luminosity feature is not available on BLUEMICROSYSTEM1).

Figure 7: Environmental page

on this page, you can choose to show the serial console to see debug information for BlueNRG transmission and characteristics subscription. These same messages are also available using a terminal for controlling the UART output (see *Figure 5: "Initial application Start"*).

Figure 8: Debug console on Environmental page
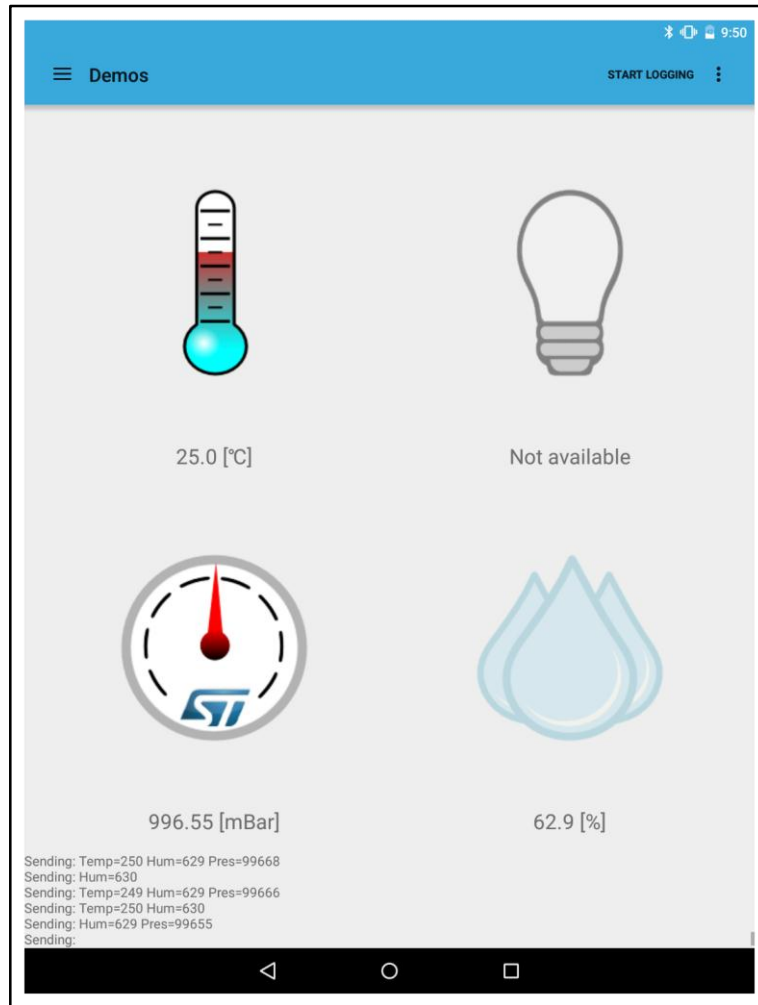
In the MEMS sensor fusion page, you will see a cube that will rotate according to board movements.

Figure 9: osxMotionFX sensor fusion page
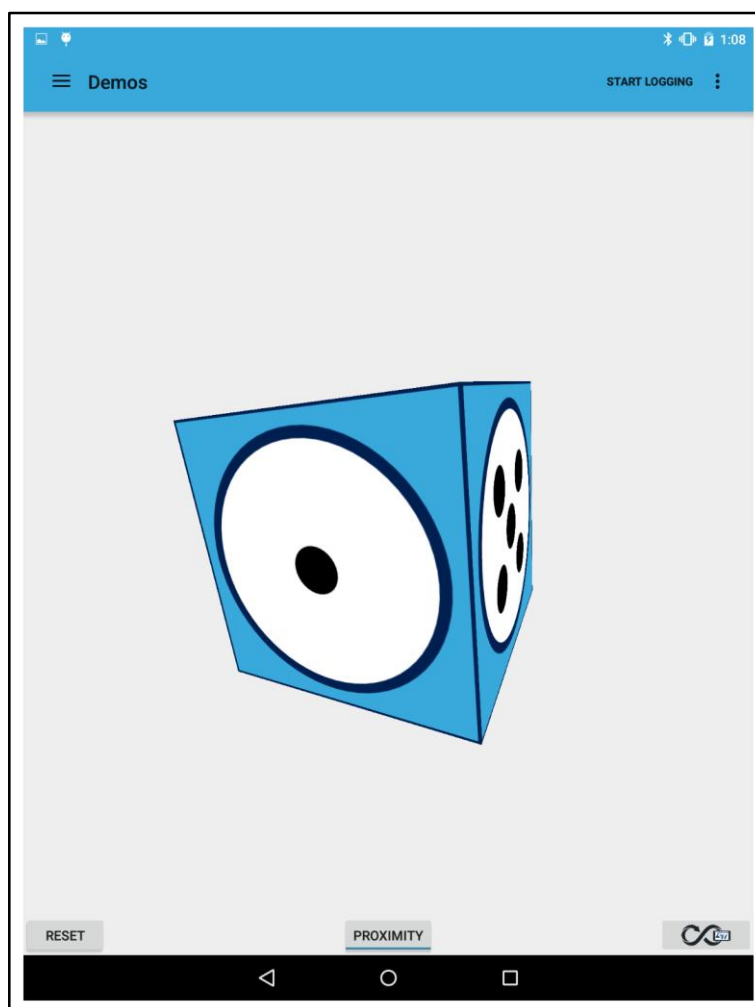
The bottom left button is for resetting the cube position; the bottom right button shows the calibration status of the osxMotionFX Library (black/green for not calibrated/calibrated) and forces a fresh magnetometer calibration when pressed. When one of these two buttons is pressed, the applications show a popup windows advising how to position the board for correct cube rotation and how to move the board to facilitate calibration.
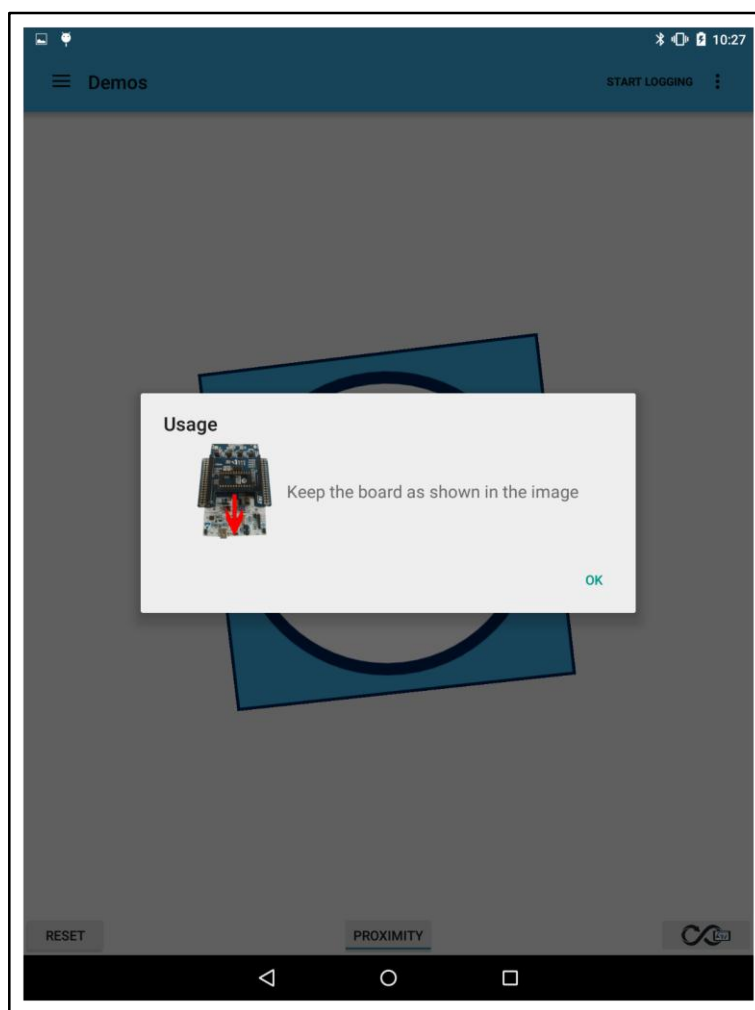
Figure 10: Left popup window

Figure 11: Right popup window

Under the default configuration, the application sends three quaternions every 30 ms. So when we move the board, the applications harmonize the sampling rate (the number of quaternions received every second) and the number of rendered frames per second that the smartphone/tablet is capable of (60 frames per second is the maximum allowed by the GPU). These two values are normally not visible, but if you touch the screen on the top left and on the top right of the display window, it is possible to show or hide these values. On the third tab, you can choose which Bluetooth characteristics to plot received value graphs for.

If the DS3 DIL 24 is present, it can also generate an interrupt when it finds a Free Fall situation.If this occurs, the Nucleo board send this information to the application, which displays a pop up window to signal this situation. The Free Fall is only enabled in the osxMotionFX page or the graph page.

Figure 12: Example of plot for osxMotionFX sensor fusion output
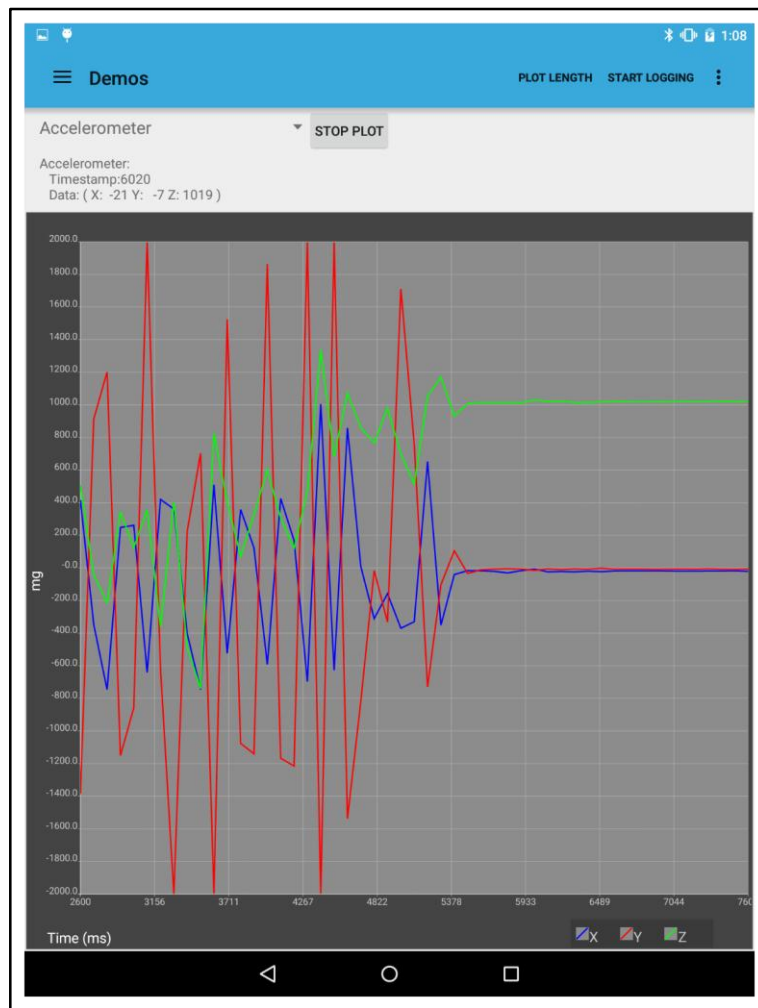
If the osxMotionAR algorithm is enabled (defining the OSX_BMS_MOTIONAR define on osx_bms_config.h) another page is visible that displays the recognized activity as one of:

- Stationary
- Walking
- Fast walking
- Jogging
- Biking
- Driving

Figure 13: osxMotionAR activity recognition page

If the osxMotionCP algorithm is enabled (defining the `OSX_BMS_MOTIONCP` define in `osx_bms_config.h`), another page is visibly that displays information about how the user is carrying the board; i.e., the phone carry position:

- on desk
- in hand
- near head
- shirt pocket
- trousers pocket
- arm swing

Figure 14: osxMotionCP carry position recognition page
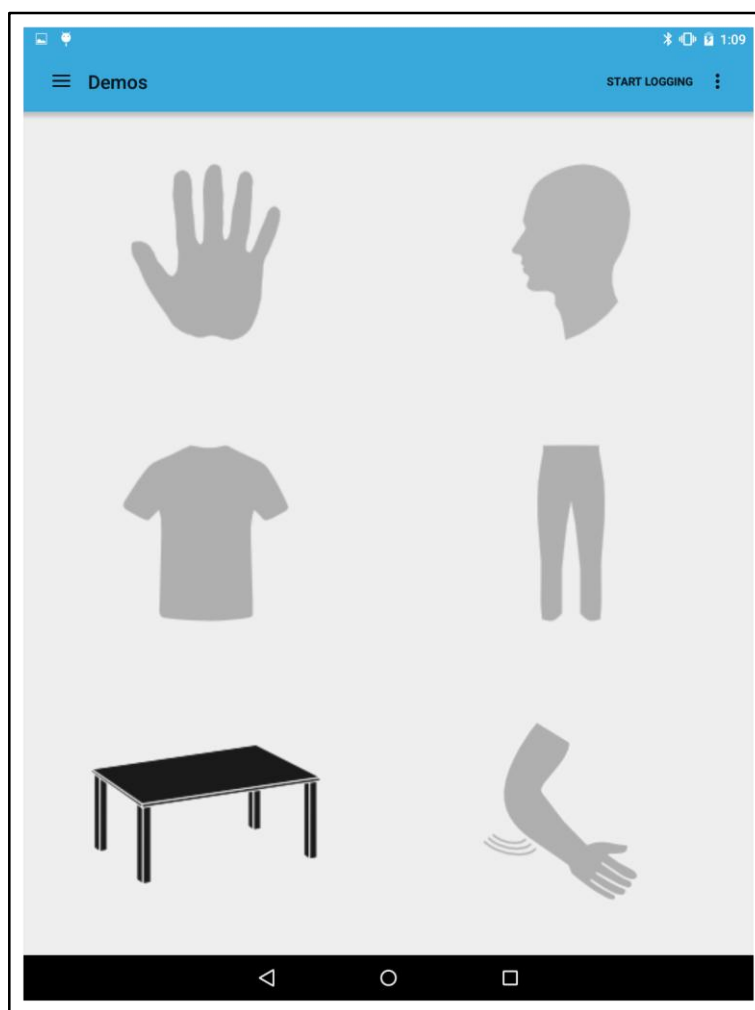
On the last tab, the application shows the Bluetooth signal strength and a grey battery (the battery service is not implemented in the BLUEMICROSYTEM1 software).

Figure 15: RSSI and battery page

In each tab, it is also possible to start/stop and save logs regarding data from the STM32 Nucleo board.

2 System setup guide

2.1 Hardware description

This section describes the hardware components needed for developing a sensors based application.

The following sub-sections describe the individual components.

The STM32 Nucleo boards provide an affordable and flexible way for users to try out new ideas and build prototypes with any STM32 microcontroller lines. The Arduino™ connectivity support and ST morpho headers make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide choice of specialized expansion boards. The STM32 Nucleo board does not require any separate probe as it integrates the ST-LINK/V2-1 debugger/programmer. The STM32 Nucleo board comes with the STM32 comprehensive software HAL library together with various packaged software examples.

Information about the STM32 Nucleo boards is available on www.st.com at

http://www.st.com/stm32nucleo

Figure 16: STM32 Nucleo board
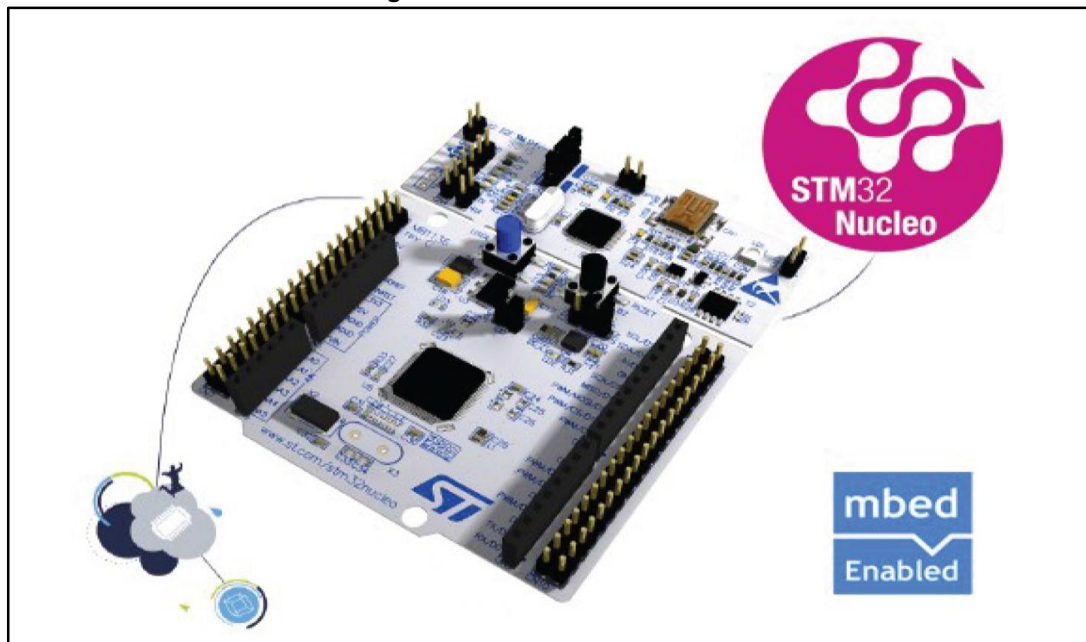
2.1.1 X-NUCLEO-IDB04A1 expansion board

The X-NUCLEO-IDB04A1 is a Bluetooth BlueNRG expansion board for use with the STM32 Nucleo system. The BlueNRG is a very low power Bluetooth Low Energy (BLE) single-mode network processor, compliant with Bluetooth specification core 4.0.

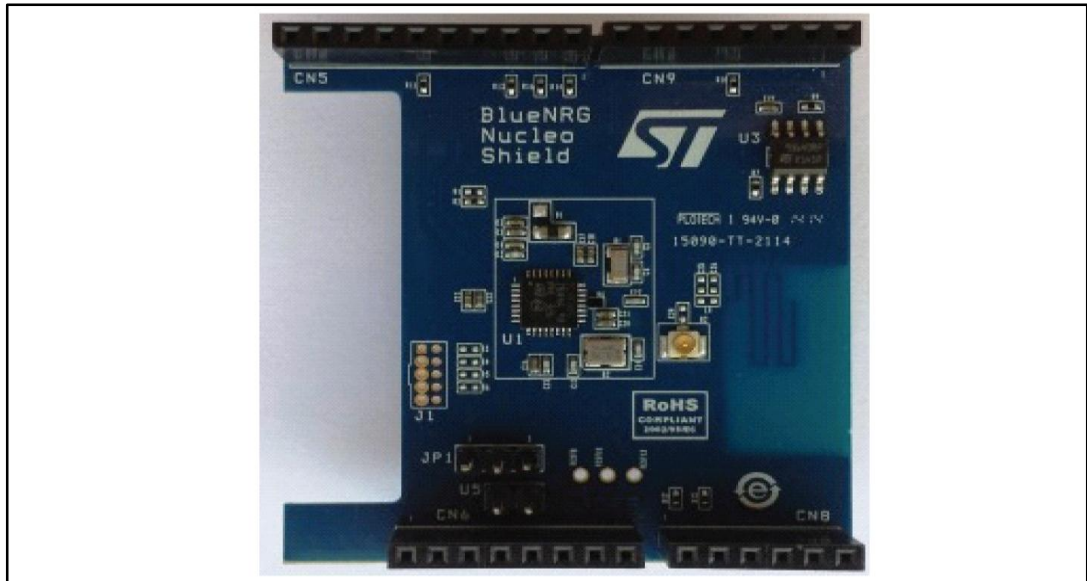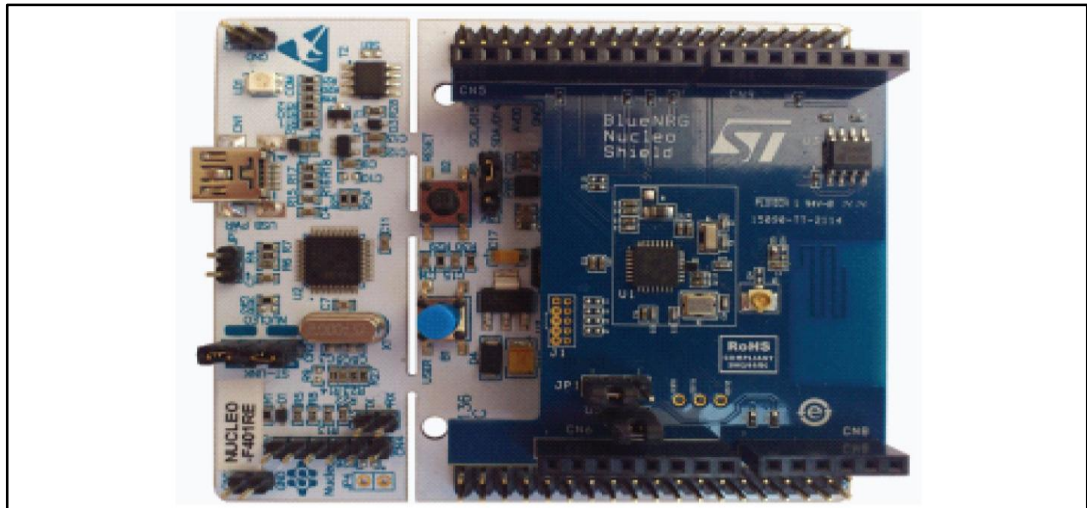**Figure 17: X-NUCLEO-IDB041 BLE expansion board**

Figure 18: BlueNRG expansion board connected to STM32 Nucleo board

Information about the X-NUCLEO-IDB04A1 expansion board is available on www.st.com at http://www.st.com/x-nucleo

2.1.2   X-NUCLEO-IDB04A1 expansion board

The X-NUCLEO-IDB05A1 is a Bluetooth Low Energy evaluation board based on the SPBTLE-RF BlueNRG-MS RF module to allow expansion of the STM32 Nucleo boards. The SPBTLE-RF module is FCC (FCC ID: S9NSPBTLERF) and IC certified (IC: 8976C-SPBTLERF). The BlueNRG-MS is a very low power Bluetooth Low Energy (BLE) single-mode network processor, compliant with Bluetooth specification v4.1. X-NUCLEO-IDB05A1 is compatible with the ST morpho and Arduino™ UNO R3 connector layout. This expansion board can be plugged into the Arduino UNO R3 connectors of any STM32 Nucleo board.
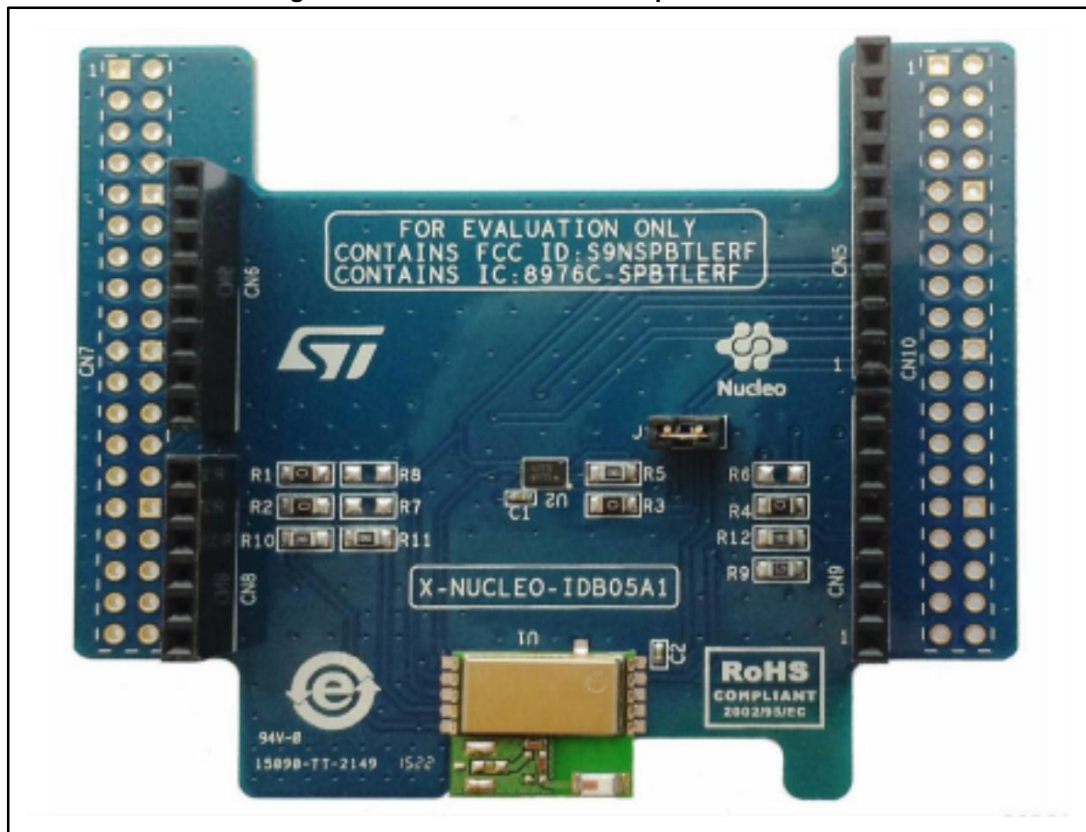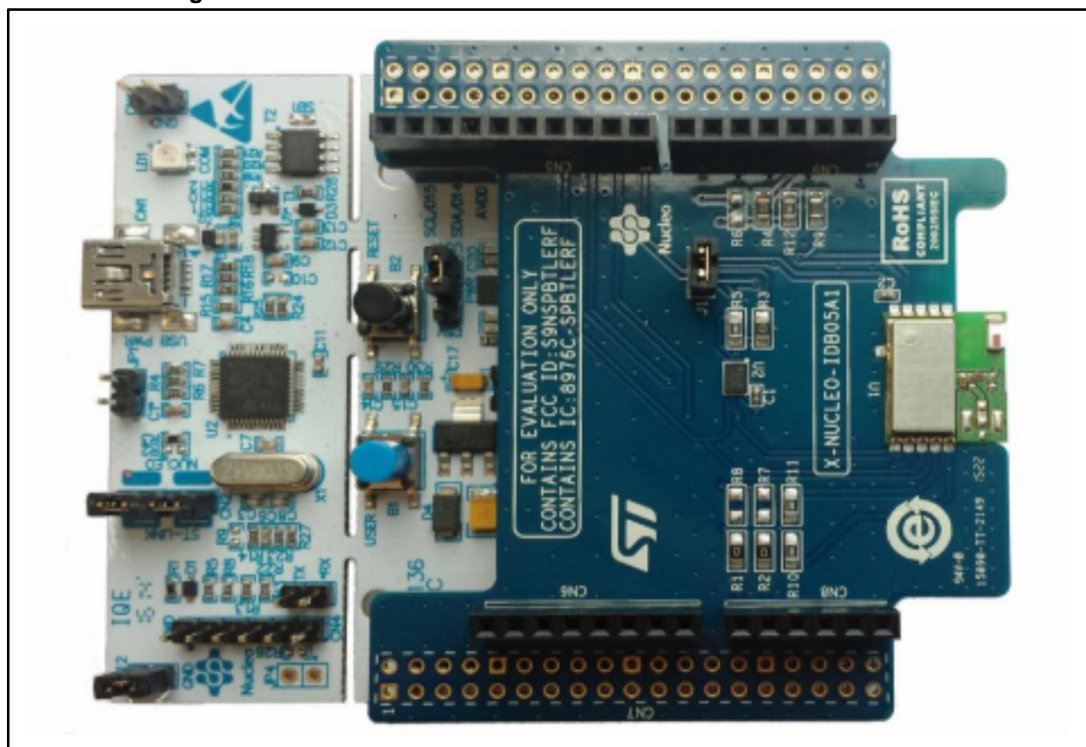
Figure 19: X-NUCLEO-IDB05A1 expansion board
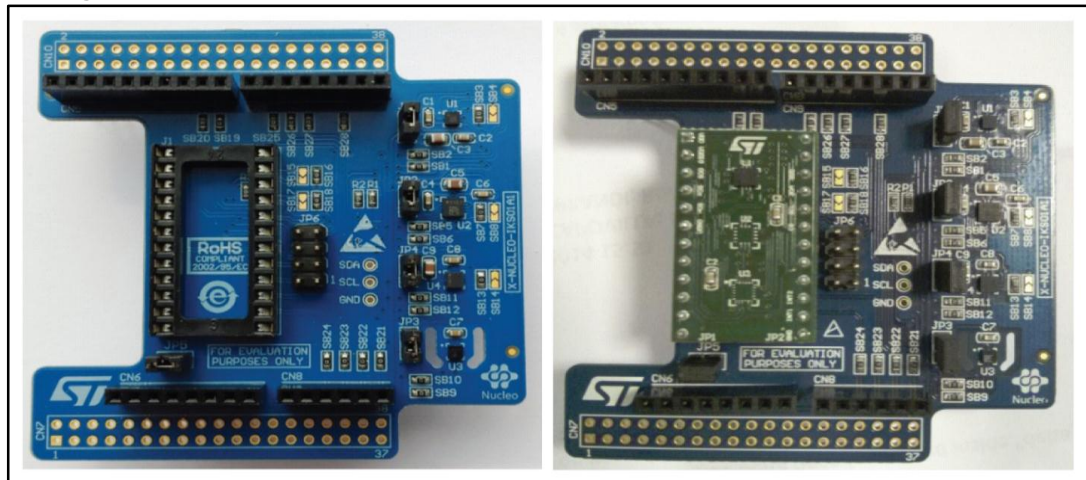

Figure 20: X-NUCLEO-IDB05A1 connected to STM32 Nucleo board

Information about the X-NUCLEO-IDB05A1 expansion board is available on www.st.com at http://www.st.com/x-nucleo

2.1.3 X-NUCLEO-IKS01A1 expansion board

The X-NUCLEO-IKS01A1 is a sensors expansion board for use with the STM32 Nucleo system. It is also compatible with Arduino UNO R3 connector layout, and is designed around STMicroelectronics humidity (HTS221), pressure (LPS25HB) and motion (LIS3MDL and LSM6DS0) sensing devices. The X-NUCLEO-IKS01A1 interfaces with the STM32 MCU via I^{2}C pin, and the user can change the default I^{2}C port and the device IRQ by changing one resistor on the evaluation board.

It's possible to attach the LSM6DS3 DIL24 expansion component and use it instead of the one for LSM6DS0.

Figure 21: X-NUCLEO-IKS01A1 sensor expansion board without/with DIL24 LSM6DS3

Information about the X-NUCLEO-IKS01A1 expansion board is available on www.st.com at http://www.st.com/x-nucleo

Figure 22: Sensor expansion boards connected to BlueNRG expansion and to STM32 Nucleo board without/with DIL24 LSM6DS3
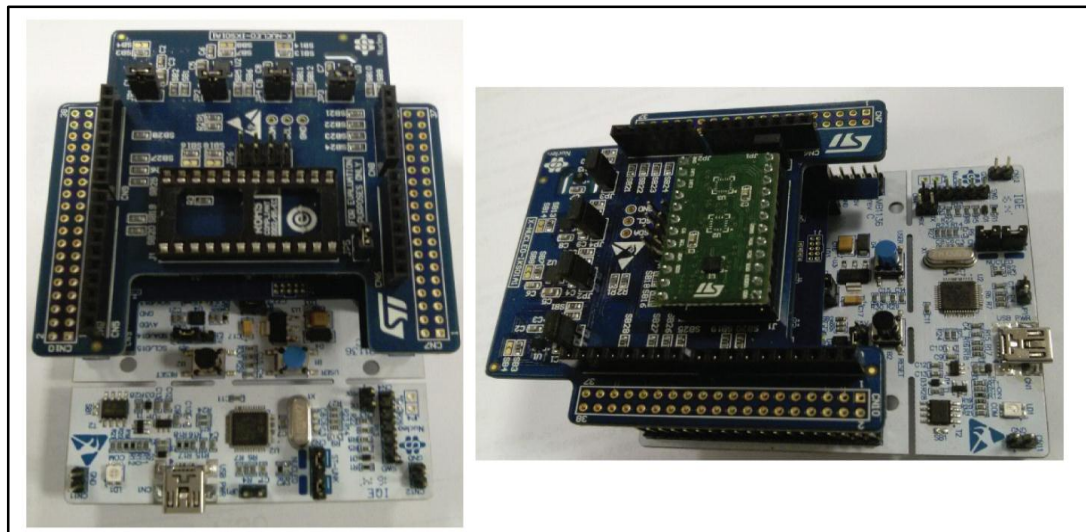
2.2   Software description

The following software components are needed in order to set up a suitable development environment for creating applications for the STM32 Nucleo equipped with the sensor and BlueNRG expansion boards:

- BLUEMICROSYSTEM1: Bluetooth Low Energy and sensor software package for STM32Cube. The BLUEMICROSYSTEM1 firmware and related documentation is available on www.st.com.
- Development tool-chain and Compiler: The STM32Cube expansion software supports the following three environments:
    - IAR Embedded Workbench for ARM® (EWARM) toolchain + ST-Link
    - RealView Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
    - System Workbench for STM32 + ST-LINK

2.3   Hardware and software setup

This section describes the hardware and software setup procedures and the system setup required.

2.3.1   Hardware setup

The following hardware components are required:

1. One STM32 Nucleo Development platform (order code: NUCLEO-F401RE or NUCLEO-L476RG)
2. One Sensor expansion board (order code: X-NUCLEO-IKS01A1)
3. One BlueNRG Bluetooth low energy expansion board (order code: X-NUCLEO-IDB04A1 or X-NUCLEO-IDB05A1)
4. One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC

2.3.2   Software setup

This section lists the minimum requirements for developers to set up the SDK, run the sample testing scenario based on the GUI utility and customize applications.

2.3.3   Development Tool-chains and compilers

Please select one of the integrated development environments supported by the STM32Cube expansion software, and read the system requirements and setup information provided by the selected IDE provider.

2.3.4   System setup guide

This section describes how to set up different hardware parts before writing and executing an application on the STM32 Nucleo board with the sensor expansion board.

2.3.5   STM32 Nucleo and sensor expansion board setup

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. The developer can download the relevant version of the ST-LINK/V2-1 USB driver by searching for STSW-LINK008 or STSW-LINK009 on www.st.com (based on Microsoft Windows OS).

The X-NUCLEO-IDB04A1/X-NUCLEO-IDB05A1 BlueNRG Bluetooth expansion board can be easily connected to the STM32 Nucleo motherboard via the Arduino UNO R3 extension connector, see *Figure 18: "BlueNRG expansion board connected to STM32 Nucleo board"*.
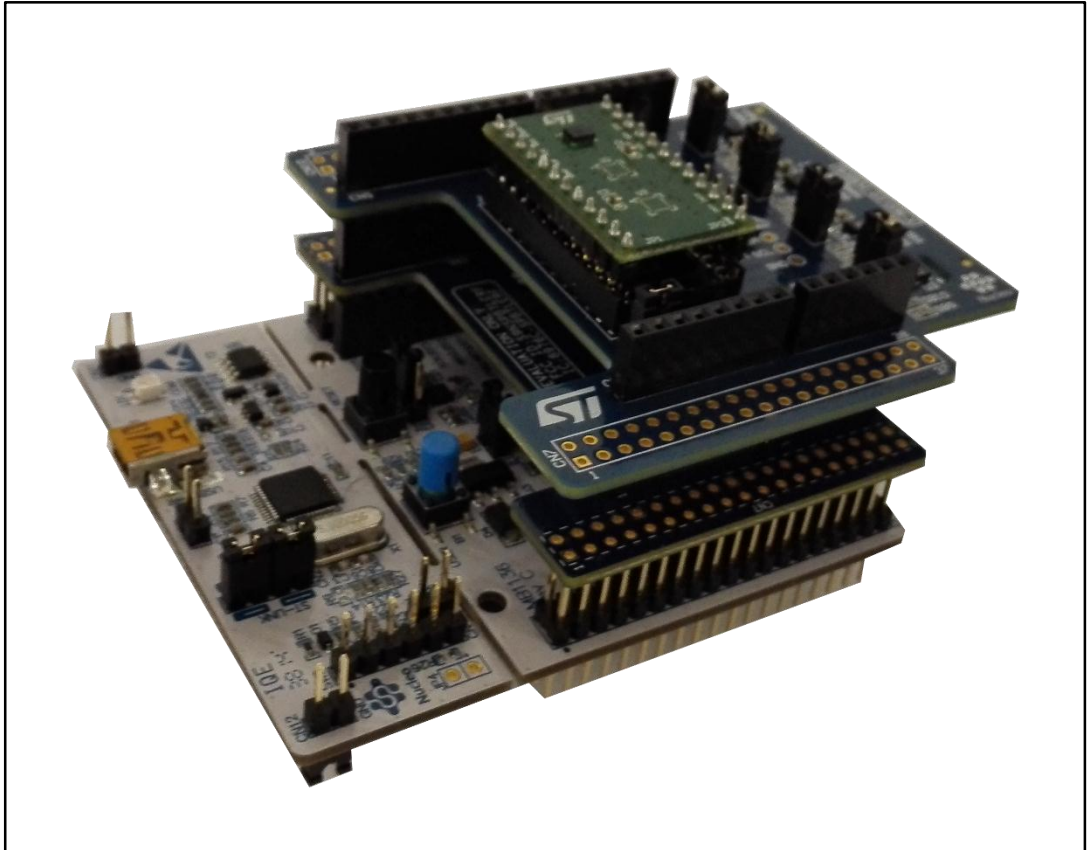
The X-NUCLEO-IKS01A1 Sensor expansion board can be easily connected to the X-NUCLEO-IDB04A1/X-NUCLEO-IDB05A1 expansion board through the Arduino UNO R3

extension connector, see *Figure 22: "Sensor expansion boards connected to BlueNRG expansion and to STM32 Nucleo board without/with DIL24 LSM6DS3"*.

Stack the boards in the following sequence to optimize the performance of the SPBTLE-RF module present on the X-NUCLEO-IDB05A1 expansion board and to reduce antenna interference (see figure below):

1.   NUCLEO-F401RE/NUCLEO-L476RG
2.   X-NUCLEO-IDB05A1
3.   X-NUCLEO-IKS01A1

Figure 23: Board stacking arrangement

3 Revision history

Table 1: Document revision history

Date	Revision	Changes
04-Mar-2015	1	Initial release.
29-Jul-2015	2	Updated Section 1.6.
13-Nov-2015	3	Throughout document:
		- minor text and image edits
		- added reference to osxMotionAR activity and osxMotionCP carry position recognition software
		- all references to X-NUCLEO-IDB04A1 changed to X-NUCLEO-IDB04A1/X-NUCLEO-IDB05A1
		Updated *Section 1.5: "Sample application description"*
		Updated *Section 1.6: "Android and iOS sample client application"*
		Added *Section 2.1.1: "X-NUCLEO-IDB04A1 expansion board"*
		Updated *Section 2.3.5: "STM32 Nucleo and Sensor expansion boards setup"*

IMPORTANT NOTICE – PLEASE READ CAREFULLY